Joint Mapping Toolkit – Visualization (JMV)

Peter Kunkel

Agenda

- Overview
- Architecture
- APIs

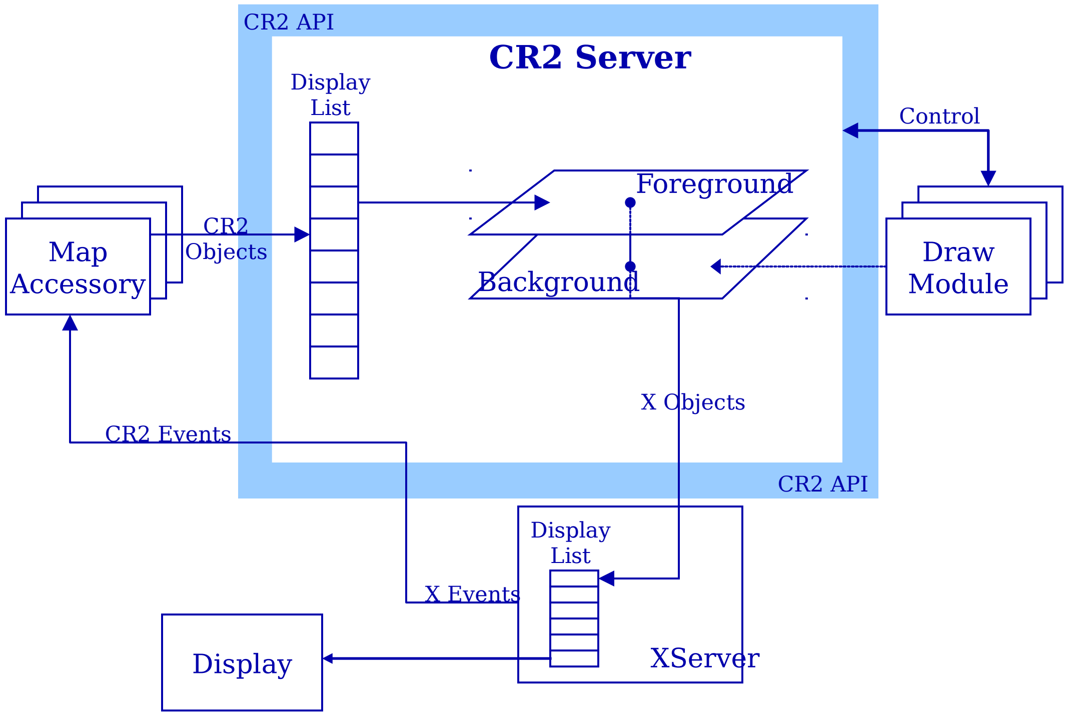
JMV Overview

- JMV provides the framework under which multiple applications can share a single map to create a complex composite picture.
- The composite picture consists of two distinct conceptual layers
  - Background Map
  - Foreground Objects.

JMV Overview (2)

- *Background Map*
  - *Created by plug-in components known as Draw Modules.*
  - *Draw Modules contribute to the background map by*
    - *Rendering a map product s such as CADRG*
    - *Drawing particular geographic features such as roads or navigation aids.*
- *Foreground Objects*
  - *Graphic objects that are drawn in layers on top of the background by the server.*
  - *Representative graphic objects are arcs, circles, corridors, boxes and military symbols (e.g., MIL-STD-2525B).*
  - *Created by Map Accessory clients*

JMV Architecture



The CR2 Server allows multiple Map Accessories to share a common geographic display.

JMV APIs

- *Zm is the C language API to Cartographer and performs the following three functions:*
  - *Zm provides the inter-Client protocol of requests and replies between Cartographer and a connected Cartographer Client.*
  - *Zm orchestrates methods to set and get elements of Zm opaque data types.*
  - *Zm permits the creation and manipulation of Display List objects.*

Map Accessory Development

- *Map Accessories follow the following design template:*
  - *Connecting to Cartographer*
  - *Attaching to a Map Window*
  - *Creating Map Objects*
  - *Handling events*
- *Sample Code Listings*
  - *C Example*
  - *Java Example*

Java Examples

- *Connecting to Cartographer*

```
public static void main( String args[] )
     { IJmvMapApp app = new JmvMapApp(args);
```

- *Attaching to a Map Window*

```
       IJmvMapWin win = app.getMapByName("System");
```

Java Examples (2) – Creating Map Objects

```
JmvGraphicsModel model = new JmvGraphicsModel();
model.setForeground(color);
model.setBackground(color);
model.setTextColor(new JmvPenColor(255, 255, 255));

int font = m_win.getFont("Courier");
model.setFont(font);
model.setFillType(JmvGraphicsModel.FILL_STIPPLED);

JmvCircle circle = new JmvCircle(info.getPosition(), radius);
IJmvPoint textPoint = new JmvPoint(info.getPosition().getLat(),
        info.getPosition().getLng() + 1.0);
JmvText text = new JmvText(textPoint, info.getName());

circle.setGraphicsModel(graphicsModel);
circle.setAutoAnimate(false);
text.setGraphicsModel(graphicsModel);
text.setAutoAnimate(false);

m_win.addObject(circle);
m_win.addObject(text);ublic static void main( String args[] )
{ IJmvMapApp app = new JmvMapApp(args);
```

Java Examples (3) – Handling Events

```
      m_win.addChartObjectListener(new JmvChartObjectListener() {
      IJmvPoint m_pnt1;
      JmvLineAnimation m_animLine;
      public void objectDown( JmvObjectEvent e )
      {
            JmvChartObject obj = e.getObject();
            if (obj instanceof JmvCircle) {
                  JmvCircle circ = (JmvCircle)obj;
                  if (m_pnt1 == null) {
                         m_pnt1 = circ.getCenter();
                         m_animLine = new JmvLineAnimation( m_pnt1, m_pnt1 );
                         m_animLine.setAnimationType((int)JmvLine.LINE_PNT2);
                         m_win.addObject(m_animLine);
                  }
                  else { m_win.removeObject(m_animLine);
                         JmvLine line = new JmvLine(m_pnt1, circ.getCenter());
                         line.getGraphicsModel().setPickableTgl(false);
                         line.setAutoAnimate(false);
                         m_win.addObject(line);
                         m_pnt1 = null;
                  }
            }
      }
});
```